

(1)

"Synchronization"

pc العادي يستخدم clock puls (rising أو falling)
سرعة computer مرتبطة بسرع الممارات الداخلية أي كلما كان clock أسرع
أسرع يكون الخطأ في signal أسرع

* لقصد به المعجزة التي تنقل مع بعضها

المحتوى
محتاج إليه - الزمن - زمن المعجزة - حتى ينفذ أحد أفرادها ولا standard.

why it needed

① client to server أو peer to peer (أي في كلا الاتجاهين)
server و وضع آخر client

⇒ time, ack, TCP msgs " في ترتيب زمني
المبادلة.

② scheduling tasks → multiprocessors

دور من يحصل مشاكل في انما الجميع يقرأ الresources ولكن المشكلة في حالة احدهم data.
يلزم وقتا على Check عليها.

③ adjusting time ≅ synchronization

في حالة انه 2 client مختلفين في الوقت حيث يمكن ان يعمل drop data في حين
أخر متأخر فيظهر انه متأخر في الترخيص في حين انه الأصح.

لا يمكن ان تكون تلك الأجهزة في العالم ان تصبح synchronized.

يحتاج precedence order الترتيب الزمني لترتيب الTasks الموجودة

Computer logical time

كل computer يحتوي على كريسالة قسوى عدمه oscillators. يتحدد الزمن
وآدم مع مرور الوقت عند تلف.

(2)

كل oscillation يعمل increment counter وكل 1 يصبح counter = 0 يزداد الزمن كما
by 1

حيث يعمل interrupt ويعمل reloading للفترة $C = 1000$ ويبدأ العمل مرة أخرى
ويعمل كل $Clock + Tick$

1 داخل ال Time في بداية Booting يعتبر مسئوله user

clock هو الوقت الذي احتاجه كل tick التي تأتي بعدها كما إذا كانت
يزداد 1 Sec.

clock skew $\leftarrow (\pm)$ الفرق في الوقت بين كل client 2 "الساعة لا تتفق" وبعدها
وخلوها

(1) عند حدوث وجود علاقة بين Time وبعضها

"happen Before" "من كل event قبل منه"

لو عندنا 2 tasks a, b و a يتنفذ قبل b وهذا ترتيب tasks لا يمكن أن يتواءم مع
كل ترتيب دل على أنه a تنفذ قبل b

Lamport logical clock

أي لا يحتاج إلى ساعة كادى وآلة من على Communication بين الآلة
حيث على Synchroniz بينهم أي يظبطوا الوقت مع بعض
بحيث يصبح sequence يتألف من logical

Time من يتفق يكون بال آلة -

$$e < e' \Rightarrow C(e) < C(e')$$

clock.

(3)

8
10
9

هناك فكر آخر حيث انه الدقيق هو العمل
ولكنه قليل مستحب.

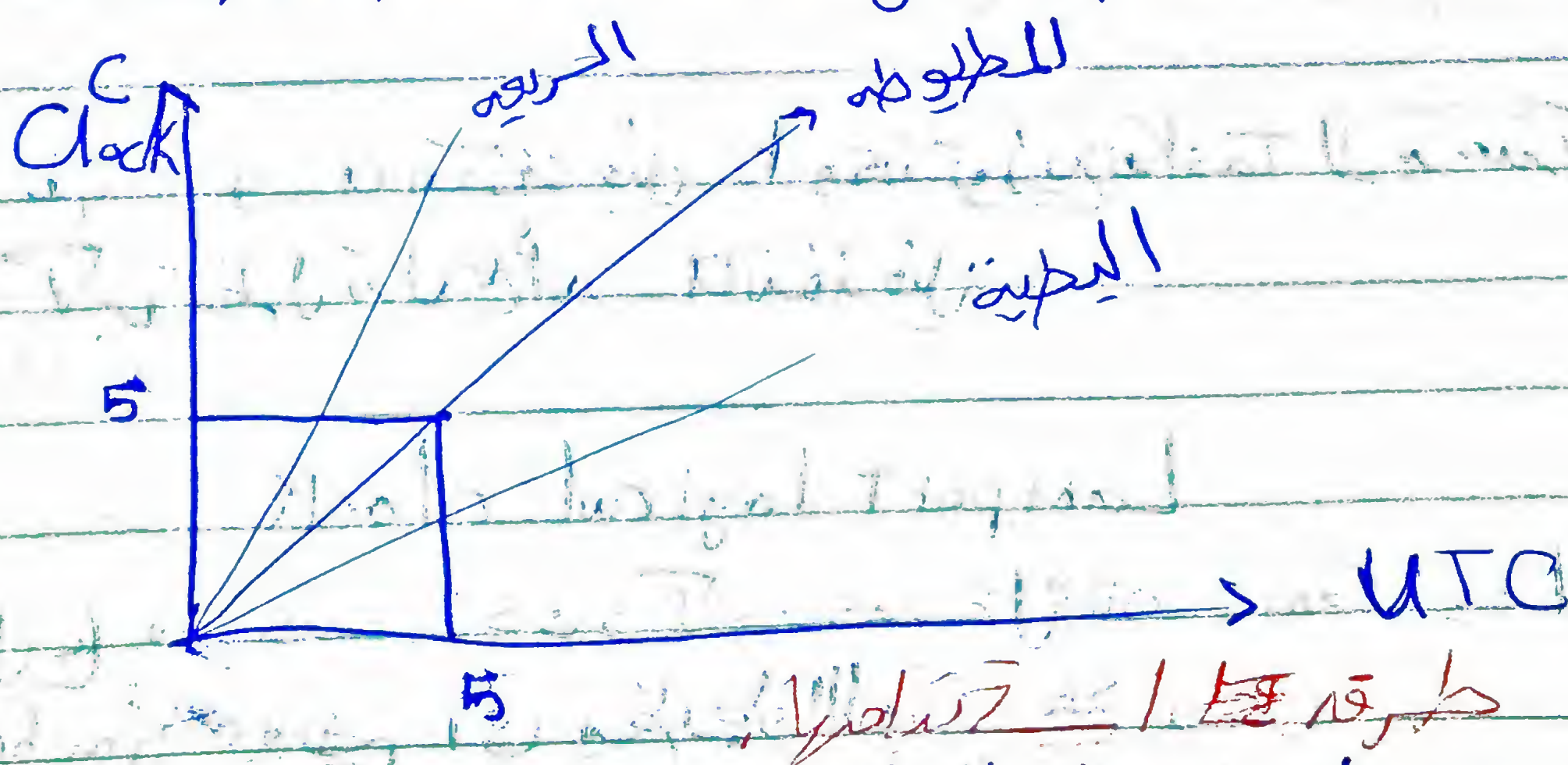
"physical clock synchronization"

- حيث يتواجد هناك ساعة كونية مظلومة، رغم أي تغيرات.
- ولكنه مع مرور الوقت قلت سرعته دورانه الأرض حيث أصبحت تأخوفاً لاهل.
- ولذلك تم ضبطها على سرعة الشمس على مدار اليوم.

Atomic clock

- ساعة دولية متواجده في أمريكا المتحدة (Universal Coordinated time)
- ولا فترة يتم تبديلها 1 sec
- "Solar hour"

- هناك هناك للناس ويتم عمل ^{casting} Broad ^{Testing} العالم على سرعة الأقمار الصناعية.



(1) Time server هو الساعات Time ونظير ذلك

"Algorithm"

- Time server مرتبط UTC كل فترة يعمل interrupt ويربط نفسه مع UTC
- "اما على طريق Satellites أو على طريق website"
- وبعد ذلك Client يصل access server ليصل نفسه

(2)

كل oscillation يتكامل Decrement Counter وكل لا يصبح $Counter = 0$ يزود الزمن $by 1$

حيث يعمل interrupt ويعمل reloading للفترة $C = 1000$ ويبدأ العملية مرة أخرى
clock tick

1. خلال ال Time في بداية Booting يعتبر مسئوله user

clock s.w. يعتمد على الوقت التي احسب خليه و ticks التي تأتي بعد حاجتها كإشارة
يزود 1 Sec.

$clock skew \leftarrow (+)$ الفرق في الوقت بين كل client 2 الساعات الأخيرة
وخلوها

(1) عند ختم وجود خلافة في Time وبعضها
happen Before " من كل event قبل منه

لو عندنا task 2 a, b و a يتنفذ قبل b وكذا توزيع tasks لا يمكن ان تكون
كل نظرنا على a تنفذ قبل b

Lamport logical clock

أي لا حيزا يتخلل ساعة كادى و لكن على Communication بين الاجزاء
حيث على Synchronizing أي يظبطوا الوقت مع بعض
يحيث يصبح sequence بيانات logical

Time من يقع يكون بال -

$$e < e' \Rightarrow c(e) < c(e')$$

clock.

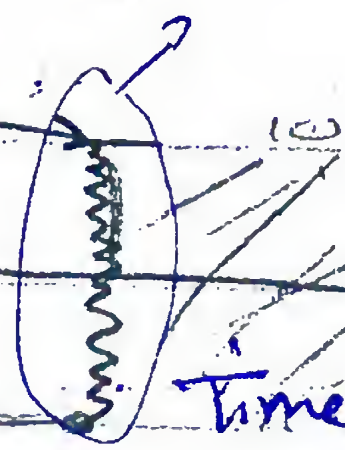
(4)

$$\frac{T_1 - T_0}{2}$$

T_0

T_1

الزمن هنا



(1)

RTT

Round Trip Time

Time server

example

T_0
10:00

10:02

T_1
10:03

$$\text{Round time} = \frac{T_1 - T_0}{2}$$

من رايح اوزمه الجاي

$$= \frac{3}{2} = 1.5$$

$$\text{New time of the machine} = \text{UTC time} + \text{Round Time} = 10:02 + 00:01.5$$

$$= 10:03.5$$

التيار انه يتغير من بعض الاغراض وقتنا من غير حصة
هذه اذ اننا نضرب الذهاب في طريق العودة
والزمن الذي فيه في الذهاب = زمن العودة
في حين وجود اثنان مختلفين

② UNIX algorithm

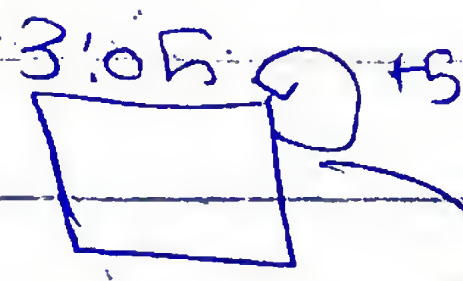
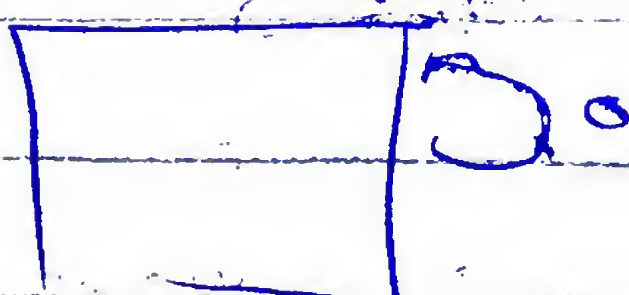
deamon سيطر
وتنظم الوقت به اذ يسرع

هو هنا هو الى ينادى في الاجهزة كيف نظمت نفسها

يجمع الفروق ولدها ينقسم في كدهم الى (average time) ولدها ينقسم

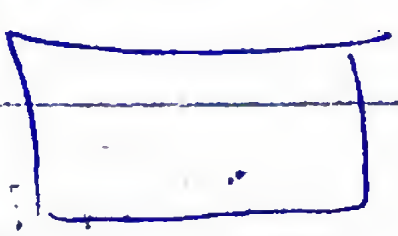
$$0 + 25 - 10 = 15$$

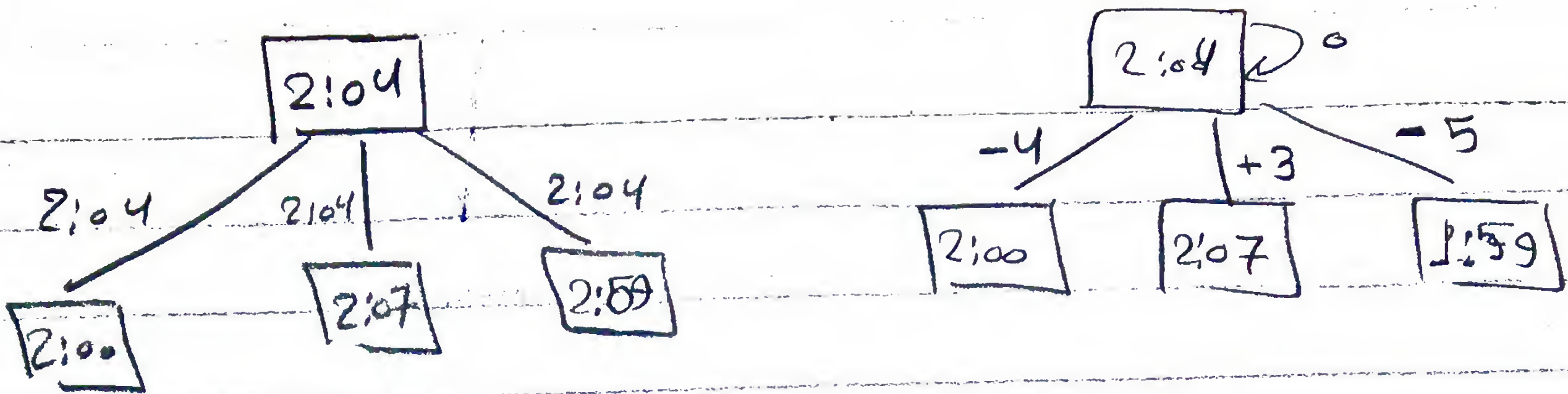
$$\frac{15}{3} = 5$$



$$-20 = 5 - 25$$

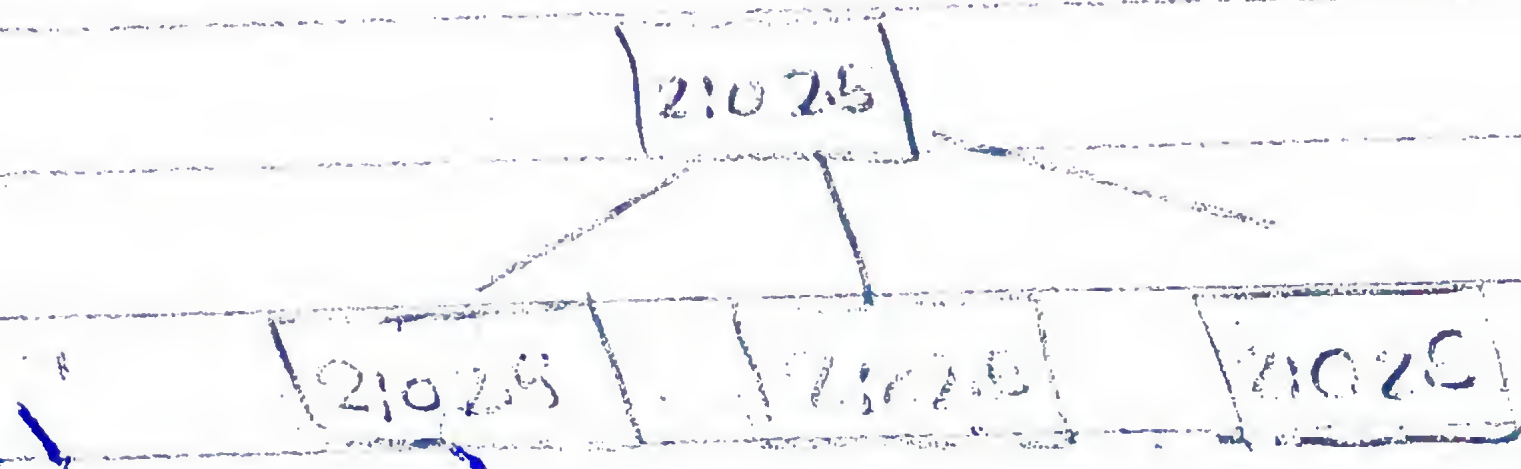
$$5 - (-10) = 15$$





$$\text{average} = \frac{-4 + 3 - 5 + 0}{4} = -1.5$$

error



$$\text{New time} = \text{old time} + \text{average error}$$

decentralized average algorithm

كل فترة يعمل على التجميع ويصبح كل حيز مسؤولاً عن نفسه حيث
يعمل Broadcast ويستوفى الفروقات ويبلغ نفسه ويبلغ نفسه لزمرة
والزمان يعمل زياً من حيز يخطط نفسه كمرحور

لذلك يمكن أن تكون هناك Single machine

يتكرر period time والأجهزة التي تتلقى تردد يتم على يد هم وقت
والزمن حيز يرد هو حور
التي هي flexibility

في حالة وجود فروق كبيرة وصغيرة حيزاً على يد تطاهلهم وحساب المتوسط
على الباقين

Round Trip → قد يكون ثابت أو متغير
أو يكون مختلفاً بين حيزاً